

420-220-SF

ALGORITHMIQUE ET PROGRAMMATION II

TRAVAIL PRATIQUE #3

MASTERMIND

PONDÉRATION : 20 POINTS

TRAVAIL EN ÉQUIPES DE DEUX OBLIGATOIREMENT

OBJECTIFS PÉDAGOGIQUES

Ce travail vise à développer les capacités suivantes:

- Le développement d'une application de moyenne envergure
- L'intégration de plusieurs notions en un même projet
- L'utilisation adéquate des structures de données
- La recherche autonome d'information
- Le développement d'algorithmes complexes
- L'application des standards de conception, de documentation et de programmation

MANDAT

Implémenter une librairie statique servant de moteur (traitement des calculs) pour une application.

Produire la documentation pertinente de l'application.

SPÉCIFICATIONS DE L'APPLICATION

DESCRIPTION GÉNÉRALE

Le Mastermind est un jeu de déductions où un joueur doit analyser des combinaisons de couleurs possibles en fonction de contraintes qui lui sont soumises. Voici les règlements de la version du jeu Mastermind que nous suivrons pour ce travail pratique.

Le jeu Mastermind, mode débutant:

Le déroulement du jeu est simple. Avant de démarrer la partie, le maître du jeu doit choisir une séquence secrète de 4 couleurs (parmi 8 proposées) et dévoiler de l'information sur cette dernière à chaque tour.

Le joueur doit essayer de deviner la séquence de 4 couleurs que le maître du jeu a choisie. Pour ce faire, le joueur doit proposer au maître du jeu une séquence de 4 couleurs qu'il a lui-même choisie.

Le maître du jeu doit ensuite dévoiler des indices sur la séquence secrète en fonction de la séquence proposée par le joueur.

Les indices suivants sont délivrées couleur par couleur à partir de la séquence proposée par le joueur:

- 1) La couleur est présente dans la séquence secrète et est à la bonne place dans la séquence proposée;
- 2) La couleur est présente dans la séquence secrète mais n'est pas à la bonne place dans la séquence proposée;
- 3) La couleur n'est pas présente dans la séquence secrète.

Le joueur propose une nouvelle séquence en fonction des indices dévoilés. Le joueur dispose de 8 essais pour trouver la séquence secrète.

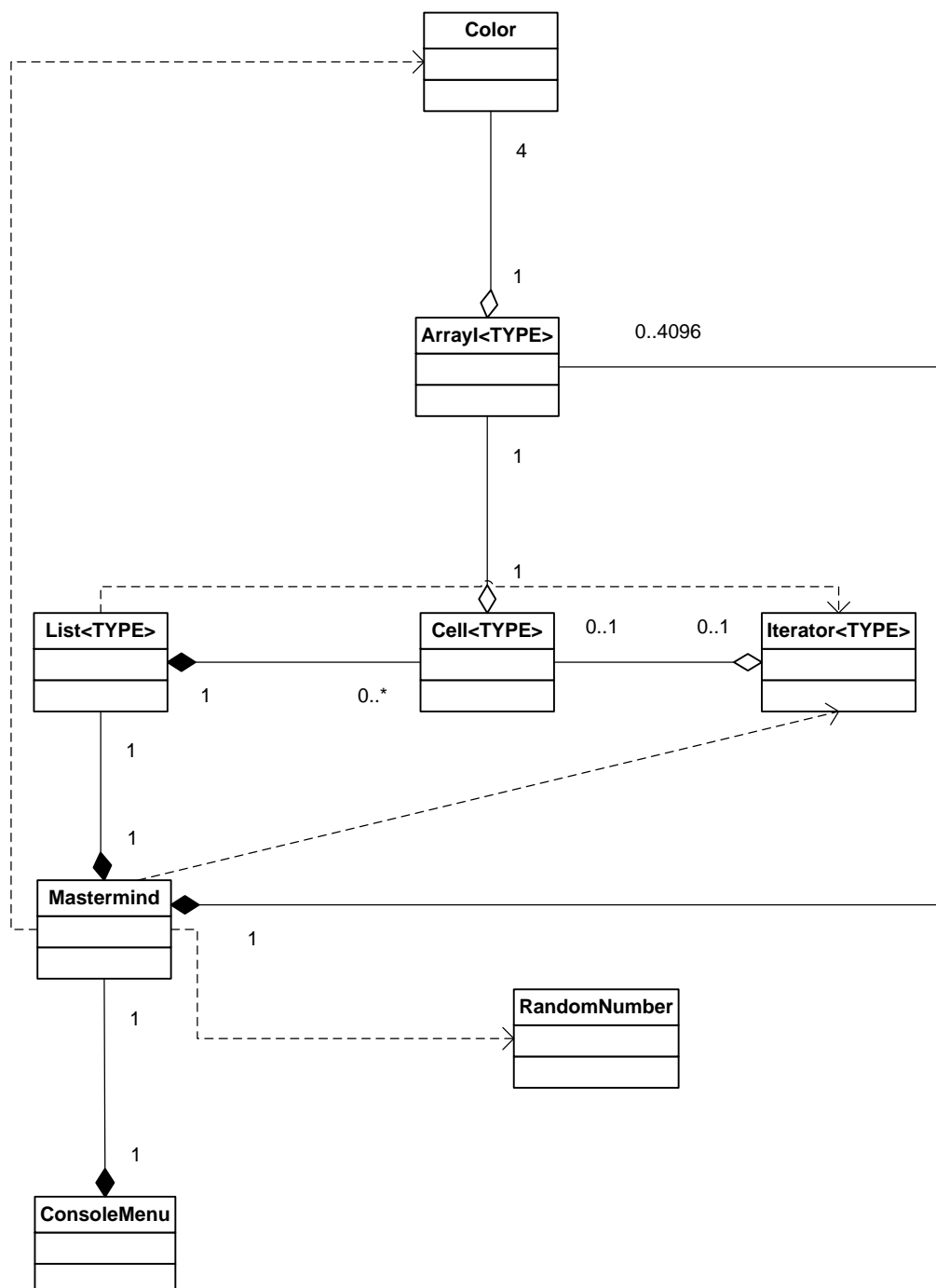
Précision sur l'implémentation du logiciel à développer:

Le jeu Mastermind que vous devez développer doit mettre en scène un être humain dans le rôle du maître du jeu et l'ordinateur dans le rôle du joueur qui essaie de deviner la séquence cachée.

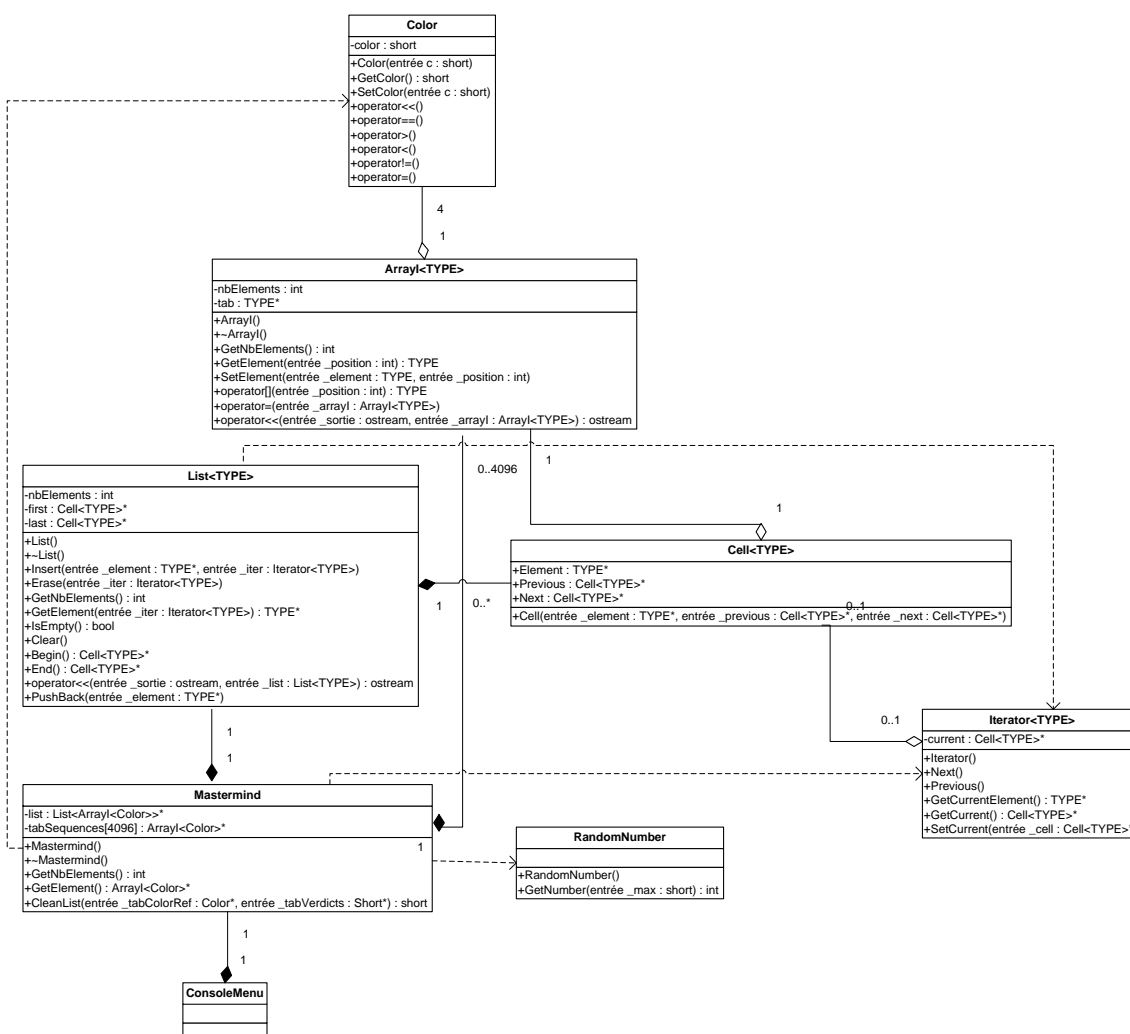
Consulter les exécutables **MastermindC.exe** (version console) et **MastermindI.exe** (version comprenant une interface programmée en C#) pour prendre connaissance du logiciel que vous devez programmer.

Voici le diagramme de classes à implémenter :

« Version sans les attributs et méthodes » pour une compréhension générale :



« Version avec les attributs et méthodes » pour une compréhension complète :



Votre mandat va se dérouler en deux étapes.

1^{ère} étape :

Vous devez programmer les classes `List<TYPE>`, `Cell<TYPE>` et `Iterator<TYPE>`. **Toutes les méthodes** qui sont présentes dans le projet **MastermindLibs/List** doivent être implémentées. De plus, le projet **MastermindLibs/TestList** doit générer un **programme exécutable** qui démontre que toutes les fonctionnalités de votre librairie ont été testées. Pour implémenter ce programme, vous devez reproduire un comportement similaire au projet « **TestUnitairePile** » des notes de cours de la semaine 8 sur Léa.

Suggestion : Afin de développer de bonnes manières de programmation, vous devriez toujours commencer par programmer une méthode de test avant d'implémenter une méthode d'une classe. **En quelque sorte, le développement des tests doit dicter le développement de votre classe à implémenter, c'est ce qu'on appelle le TDD (Test Driven Development).**

Date de la remise de la première étape : **au plus tard mercredi le 30 avril avant minuit.**

2^{ème} étape :

Vous devez :

- 1) Programmer la classe `Mastermind` (projet `MastermindLibs/Mastermind`) en respectant obligatoirement la représentation de la classe `Mastermind` dans le diagramme de classes.

- 2) Programmer la classe ConsoleMenu (projet MastermindC++). Cette classe doit s'approprier le contenu du « main » et le redécouper de façon élégante. Vous pouvez vous inspirer des classes ConsoleMenu que vous avez implémentées lors de vos deux premiers travaux pratiques. Attention, votre classe ConsoleMenu doit offrir à l'utilisateur le même comportement que ce qu'offre le « main »; pour l'utilisateur, ce changement doit être transparent.

Date de la remise de la deuxième étape: **au plus tard mercredi le 14 mai avant minuit.**

ÉVALUATIONS

La première remise sera évaluée selon les critères suivants :

Éléments	Pondération
Conception et réalisation des tests unitaires (MastermindLibs/TestList)	30%
Fiabilité du code (respect des spécifications pour les classes List<TYPE>, Cell<TYPE> et Iterator<TYPE>)	30%
Qualité du code (optimisation, clareté)	15%
Documentation du code source (interne et externe)	10%
Respect des normes de programmation	10%
Feuille de temps	5%

La première remise compte pour 40% de la note de ce travail pratique.

La deuxième remise sera évaluée selon les critères suivants :

Éléments	Pondération
Fiabilité du code (respect des spécifications de l'application)	50%
Qualité du code (optimisation, clareté, découpage du projet MastermindC++ à l'aide de la classe ConsoleMenu)	30%
Documentation du code source (interne et externe)	10%
Respect des normes de programmation	5%
Feuille de temps	5%

La deuxième remise compte pour 60% de la note de ce travail pratique.

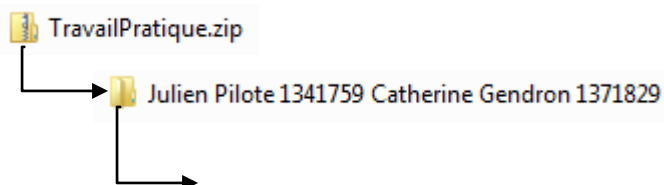
Conformément à la politique concernant la qualité du français écrit, le travail est sujet à une pénalité pouvant aller jusqu'à 20 % du total des points attribuables. Se référer à la section « **Évaluation** » du plan de cours pour plus de détails sur l'évaluation des travaux.

Si le professeur le juge à propos, tout étudiant(e) pourra être convoqué(e) à une rencontre d'évaluation pour vérifier son degré d'acquisition des connaissances et d'appréhension de la solution proposée.

BIENS LIVRABLES

Remise 1 :

Vous devez remettre sur LEA un fichier ZIP contenant un répertoire identifié à vos deux noms et vos deux matricules:

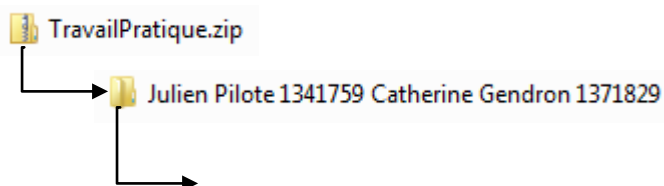


Et qui contient les répertoires suivants :

MastermindLibs	Répertoire contenant les projets Arrayl, List et TestList complétés et épurés.
doc	Documentation externe HTML (via DOXYGEN) et feuille de temps

Remise 2 :

Vous devez remettre sur LEA un fichier ZIP contenant un répertoire identifié à vos deux noms et vos deux matricules:



Et qui contient les répertoires suivants :

MastermindC++	Projet qui produit le fichier Mastermind.exe, ce projet a été découpé à l'aide de la classe ConsoleMenu. Ce projet, comme tous les autres, doit avoir été épuré.
MastermindCLR	Projet qui produit le fichier MastermindCLR.dll. Ce projet ne doit pas avoir été modifié, vous ne faites que le remettre à nouveau.
MastermindLibs	Répertoire contenant les projets Arrayl, List, TestList et Mastermind (produit le fichier Mastermind.lib utilisé par MastermindC++ et MastermindCLR) complétés et épurés (les erreurs décelées à la remise 1 à ont été corrigées.
doc	Documentation externe HTML (via DOXYGEN) et feuille de temps